

TITLE OF THE INVENTION

IMAGE FORMING APPARATUS AND CONTROL METHOD FOR THE SAME

5

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an image forming apparatus forming an image in accordance with control codes stored in a memory medium, and a rewrite control method for the image forming apparatus.

Description of the Related Art

Hitherto, memory media such as ROMs (Read Only Memories) have been employed as memory means for storing control codes used for control of image forming apparatus. Rewritable memory media such as flash ROMs have also been employed in recent years. Control codes are rewritable by transferring data to such a rewritable memory medium from an external apparatus, e.g., a host computer.

In the conventional techniques, however, a rewrite process must have been carried out for a plurality of flash ROMs for completing rewrite, and a large amount of download programs have been required to be written in the ROMs or the like. Therefore, programs are more

likely to contain bugs, and if bugs are found in ROMs or the like, those ROMs must be replaced. This has accompanied the problem of imposing a large load on rewrite work.

5 Also, the conventional techniques have had another problem in the rewrite process. Because of an interrupt area being used in common to image forming control, an interrupt area dedicated for the rewrite process cannot be secured. Accordingly, it is indispensable to monitor
10 at all times whether data is transmitted from an external apparatus, meaning that the rewrite process takes a longer period of time.

 Meanwhile, it is also known to store a rewrite program, which is necessary for carrying out rewrite of
15 control programs transmitted from an external apparatus, not in a flash ROM where information slightly tends to be unstable, but in a ROM.

 In such a case, a CPU (Central Processing Unit) executes various processes while controlling respective
20 components in accordance with the rewrite program preset in a ROM.

 Control programs are stored in a nonvolatile readable/writable ROM, allowing new control programs to be written in that ROM. On the other hand, a rewrite
25 program used for down-loading the control programs is fixedly held in a memory medium which is not updated. In

other words, changes of the rewrite program have not been taken into consideration.

As an example of the situation described above, in copying machines, printers and so on, control programs
5 have been stored in a nonvolatile EPROM (Erasable Programmable ROM) or a mask ROM, but it has become common that a flash ROM, an EEPROM (Electrically Erasable Programmable ROM), etc., which are nonvolatile memories being readable and writable on-board, are
10 employed so that the rewrite program can be updated without replacing the ROM by a new one when addition of functions or version-up of an application program will be necessary in future.

More specifically, when the rewrite program in a
15 flash ROM is updated, a new rewrite program is downloaded into the flash ROM from a host computer, a CD-ROM drive, a floppy drive or the like which is connected to an image forming apparatus.

To this end, communication means is provided in the
20 image forming apparatus, and a rewrite download program adapted for the image forming apparatus is held in a memory medium which is disposed in the apparatus and not updated.

In the conventional image forming apparatus,
25 however, because the rewrite download program is fixedly held, control programs to be down-loaded have been

restricted by the rewrite download program.

Further, because a manner of down-loading control programs is also determined by the rewrite download program, flexibility in a manner of download is small; hence it has been difficult to perform efficient download.

In the above method, for example, changing addresses of a ROM, into which control programs are to be down-loaded, has not been taken into consideration, and new control programs are down-loaded within the range of an address area originally assigned for the control programs in advance. Accordingly, if new control programs which are different in, e.g., assignment address area from the original control programs are down-loaded, the rewrite download program built in the image forming apparatus and the download application operating on the host computer cannot be utilized due to a difference in address configuration. In such a case, the ROM storing the rewrite download program must be replaced and hence the efficiency of rewrite work is lowered.

Additionally, where the rewrite download program is stored in a memory medium (e.g., a mask ROM) which is disposed in the image forming apparatus and not updated, that ROM must be replaced if there occurs any failure in the rewrite download program.

SUMMARY OF THE INVENTION

With the view of solving the problems set forth above, an object of the present invention is to provide
5 an image forming apparatus and a rewrite control method for the image forming apparatus, with which rewrite is performed with a minimum amount of program data to be stored in a ROM or the like and improved reliability by storing, in a ROM or the like, a control program just
10 required for transferring a download program to a RAM or the like, and then executing the transferred download program.

Another object of the present invention is to achieve high-speed rewrite by allowing an interrupt in
15 not only an image forming process, but also a rewrite process.

Still another object of the present invention is to efficiently down-load control programs.

Other objects of the present invention will become
20 apparent from the accompanying drawings and the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a sectional view showing a structure of
25 an image forming apparatus according to a first embodiment;

Fig. 2 is a representation showing a construction of a control panel according to the first embodiment;

Fig. 3 is a block diagram showing a construction of a control system for the image forming apparatus
5 according to the first embodiment;

Fig. 4 is a flowchart showing a data rewrite process in a flash ROM according to the first embodiment;

Fig. 5 is a flowchart showing a rewrite control
10 program transfer process according to the first embodiment;

Fig. 6 is a representation showing an example of display indicated when a flash ROM (1004) is rewritten;

Figs. 7A and 7B are each a representation showing a
15 memory map according to the first embodiment;

Fig. 8 is a block diagram showing a construction of a control system for an image forming apparatus according to a second embodiment;

Fig. 9 is a flowchart showing a data rewrite
20 process in a flash ROM according to the second embodiment;

Fig. 10 is a flowchart showing a rewrite control program transfer process according to the second embodiment;

Figs. 11A and 11B are each a representation showing
25 a memory map according to the second embodiment;

Fig. 12 is a block diagram showing a construction of a control system for an image forming apparatus according to a third embodiment;

Fig. 13 is a flowchart showing a data rewrite
5 process in a flash ROM according to the third embodiment;

Figs. 14A and 14B is a representation showing a memory map according to the third embodiment;

Fig. 15 is a flowchart showing a control program
10 write process according to a fourth embodiment;

Fig. 16 is a block diagram showing a construction of an image forming apparatus according to a fourth embodiment; and

Fig. 17 is a flowchart showing a control program
15 write process according to a fifth embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will be described below with reference to the drawings.

20 [First Embodiment]

Fig. 1 is a sectional view showing a structure of an image forming apparatus 100 according to this first embodiment. In Fig. 1, reference numeral 1 denotes a reader section and 2 denotes a printer section.

25 Constructions of the reader section 1 and the printer section 2 will be described below separately.

In the reader section 1, document sheets placed on a document feeder 101 are successively carried one by one onto a document glass plate surface 102. When the document sheet reaches a predetermined position on the document glass plate surface 102, a lamp 103 in a scanner unit 104 is lit up and the scanner unit 104 is moved to illuminate the document sheet.

Light having reelected from the document sheet is introduced to a CCD image sensor unit (hereinafter referred to as a "CCD") 109 through mirrors 105, 106, 107 and a lens 108. The light having reflected from the document sheet and entered the CCD 109 is subjected to photoelectric conversion in the CCD 109 for conversion into an electric signal which is then sent to an image processing unit 110.

The electric signal, which is connected to the printer section 2 through an external switching circuit (not shown) in the image processing unit 110, is converted into a modulated optical signal by an exposure control unit 201, the modulated optical signal being irradiated to a photosensitive member 202. A latent image is formed on the photosensitive member 202 by the irradiated light and then developed by a developing unit 203. At the timing matched with the leading end of a developed image, a sheet of transfer paper is fed from a transfer paper stack cassette 204 or 205 so that the

developed image is transferred onto the sheet of transfer paper in a transfer station 206. The transferred image is fixed to the sheet of transfer paper by a fixing unit 207, and thereafter the sheet of transfer paper is carried out to the exterior of the apparatus through a paper ejecting portion 208. The sheet of transfer paper having exited from the paper ejecting portion 208 is introduced to a sorter 220. The sorter 220 operates so as to eject sheets of transfer paper into trays 2202 - 2208 when a sorting function is effected, and to eject sheets of transfer paper into the uppermost tray 2202 as default setting when the sorting function is not effected.

A manner of forming images, which are successively read, on both surfaces of one sheet of output paper will be next described. Before reaching the paper ejecting portion 208, the sheet of transfer paper having an image fixed on its one surface by the fixing unit 207 in accordance with the above-mentioned process is reversed in travel direction thereof through a travel direction switching member 209 and then carried to a resupplied transfer paper stack portion 210. When a next document sheet is prepared, a document image is read in a similar manner to the above-mentioned process. As for a sheet of transfer paper, however, the sheet of transfer paper having the image already fixed on its one surface is fed

to the transfer station 206 from then resupplied
transfer paper stack portion 210 in a reversed state.
Thus, two successive document images can be formed
respectively on front and rear surfaces of the same
5 sheet of transfer paper.

Fig. 2 shows a construction of a control panel 300
according to the first embodiment. In Fig. 2, reference
numeral 301 denotes a display section in which operating
states of the apparatus, various messages and so on are
10 displayed. The surface of the display section 301 is
constructed of a touch panel so that the display section
301 can also serve as selection keys upon the surface
thereof being touched. Reference numeral 302 denotes a
ten-key portion comprising keys used to input numerals.
15 Reference numeral 303 denotes a start key. Upon the
start key 303 being depressed, the apparatus start the
copying operation.

Fig. 3 is a block diagram showing a construction of
a control system for the image forming apparatus 100. In
20 Fig. 3, reference numeral 1001 denotes a CPU for
carrying out basic control of the entire apparatus. The
CPU 1001 is connected, via an address bus and a data bus,
to an input/output (I/O) port 1003, a flash ROM 1004 in
which image forming control programs are written, a work
25 RAM 1005 for use in execution of processing, and a
key/display control unit 1006 for controlling display on

and key inputs from the control panel 300.

Sub 17 In addition, the CPU 1001 includes a built-in ROM 1019 in which a program for transferring a control program adapted to rewrite the flash ROM 1004, etc.

5 based on data transmitted from an external apparatus, such as a host computer (personal computer), to the RAM 1005 is written. Also, the CPU 1001 has a plurality of operating modes which can be selectively set by an operating mode setting unit 1018. During the normal
10 image forming operation, the CPU 1001 is set to the mode in which the built-in ROM 1019 is not effective.

In such a control system, the CPU 1001 executes input/output control successively in accordance with the contents of the flash ROM 1004 for carrying out the
15 image forming operation, and also receives key-input information from the control panel 300 through the key/display control unit 1006. Further, the CPU 1001 instructs the key/- display control unit 1006 to indicate the operating states of the apparatus 100, the
20 operating mode set in response to the key input, etc., thereby controlling display on the control panel 300.

The image forming apparatus 100 also includes another CPU 1002 for executing control of operations of various motors, heaters and so on. The operation control
25 of the various motors, etc. is performed in accordance with instructions from the CPU 1002. The CPU 1002 is

connected, via an address bus and a data bus, to a flash
ROM 1008 in which control programs are written, a work
RAM 1009 for use in execution of processing, and an
input/-output (I/O) port 1010. Various control motors,
5 etc. are connected to the input/output (I/O) port 1010
as needed. Additionally, the CPU 1002 executes data
communication with the CPU 1001 through a dual port RAM
1007 serving as interface means.

Further, the CPU 1001 executes data communication
10 with a CPU 1014 for converting data, such as PDL codes,
which is transmitted from the external apparatus, into
an image signal through a dual port RAM 1013 serving as
interface means. Upon receiving the PDL codes from the
external apparatus, the CPU 1014 converts the received
15 PDL codes into image data, requests image forming
processing to the CPU 1001, which is control means for
an image forming section, through the dual port RAM 1013,
and then outputs the image data to the printer section 2
in synch with control of the CPU 1001. The CPU 1014 is
20 connected, via an address bus and a data bus, to a flash
ROM 1015 in which control programs are written, a work
RAM 1016 for use in execution of processing and for
developing the PDL codes into the image data, and an
input/output (I/O) port 1017. Note that the input/output
25 port 1017 is connected to a control port through which
data, such as PDL codes, is received from the external

apparatus.

In addition, the CPU 1001 executes data communication with a not-shown external apparatus, such as a personal computer, through a serial interface (I/F) 1011. The CPU 1001 also analyzes data received from the external apparatus through the serial interface 1011, and transmits data in the apparatus 100 to the external apparatus in accordance with an analyzed result. Further, the CPU 1001 performs rewrite of data in the flash ROMs 1004, 1008, 1015 based on data received from the external apparatus.

A process for rewriting data in the flash ROMs 1004, 1008, 1015 will be described below. Fig. 4 is a flowchart showing the data rewrite process in each flash ROM according to this first embodiment. This process is performed by the CPU 1001 when a power supply is turned on in a state where the built-in ROM 1019 is made effective. More specifically, when the data rewrite process is desired, this process is started and executed as described below by operating a dip switch or the like in a state where the power supply is not turned on, thereby establishing a mode in which the built-in ROM 1019 is effective, and thereafter turning on the power supply. In this connection, Fig. 7A shows a memory map used in the normal image forming operation, i.e., a memory map used for the execution of processing in the

mode in which the built-in ROM 1019 is made not effective, and Fig. 7B shows a memory map used in the rewrite control operation, i.e., a memory map used for the execution of processing in the mode in which the built-in ROM 1019 is made effective. In other words, Fig. 7A shows a memory map used in a normal process to carry out the operation of a copying machine or a printer, and Fig. 7B shows a memory map used in a download process.

As will be seen from Figs. 7A and 7B, part of the memory area assigned to the flash ROM in the mode in which the built-in ROM 1019 is not effective, is assigned to the built-in ROM 1019 in the mode in which the built-in ROM 1019 is effective. In the illustrated example, addresses from 000000h to 010000h represent an area shared by both the modes. Further, the shared area includes in its part a normal interrupt vector area in the mode in which the built-in ROM 1019 is not effective, and a transfer/-download vector area in the mode in which the built-in ROM 1019 is effective. Stated otherwise, the same memory space is occupied by the flash ROM during the image forming operation and by the built-in ROM during the rewrite control operation, thus enabling a vector area dedicated for the rewrite control to be ensured.

The reason why part of the shared memory area is provided as the transfer/download vector area in the

illustrated example is that the memory is used upon an interrupt substantially in the same range for transfer to the RAM 1005 and download to the flash ROM, and therefore a common vector area can be used in both
5 transfer and download.

Returning to Fig. 4, when execution of the rewrite control program is started, i.e., when in this embodiment the power supply is turned on in the mode in which the built-in ROM is effective, the operator is
10 informed of the fact that the apparatus is under the rewrite process and the image forming operation cannot be effected.

First, in step S100, it is determined in accordance with information described below whether the rewrite or
15 rewrite process of each of the flash ROMs 1004, 1008, 1015 has suspended or failed. If the rewrite of the flash ROMs 1008, 1015 has suspended, information indicating the suspension is noticed from the CPUs 1002, 1014 to the CPU 1001 through the dual port RAMs 1007,
20 1013. When the rewrite is determined as having suspended in response to the above notice, this means that the control data in each flash ROM is in a state not ensured.

Therefore, the control flow proceeds to step S103 described later for executing a rewrite program transfer
25 mode. Incidentally, how to determine whether the rewrite of the flash ROM 1004 has suspended will be described

later.

On the other hand, if it is determined in step S100 that the rewrite has not suspended, the control flow proceeds to step S101 to read information in the flash ROM which is a destination of installing in an install mode. In subsequent step S102, the read version is compared with the version to be newly installed. If both the versions are in match with each other, there is no need of installing data. Therefore, the CPU indicates the end of the install mode to the operator, and also prompts the operator to turn off the power supply and to set the mode back to the mode in which the built-in ROM is not effective. However, if both the versions are not in match with each other in step S102, the control flow proceeds to step S103 to execute the rewrite program transfer mode.

In step S103, in accordance with the contents of the built-in ROM 1019, a rewrite program is transferred to the RAM 1005 based on data transmitted from the external apparatus. After that, the control flow proceeds to step S104 to execute a flash ROM rewrite mode.

Ref 2 Details of control in above step S103 will now be described with reference to a flowchart of Fig. 5. When the rewrite program transfer mode is started, the data transmitted from the external apparatus is first

Sub A2 }
received in step S1031. The control flow proceeds to
step S1032 to write the received data in the RAM 105
successively. It is then determined in step S1033
whether the received data is the last data to be
5 transferred. If the received data is the last data, the
rewrite program transfer process is ended and execution
of the flash ROM rewrite mode is started. If the
received data is not the last data, the control flow
returns to step S1031 to repeat the above process.

10 Transfer of the rewrite program is performed
through the process described above. In this connection,
the built-in ROM 1019 stores therein a control program
for transferring the rewrite program to the RAM 1005.
Comparing with the case of directly storing, in the
15 built-in ROM 1019, the rewrite program for rewriting the
image forming control programs stored in a plurality of
memory media, however, it is expected that the amount of
program data to be stored in the built-in ROM 1019 is
reduced considerably as a result of the above process of
20 this embodiment. Correspondingly, it is also expected
that a probability of bugs being present in the built-in
ROM 1019 is reduced considerably.

Next, it is determined in step S104 which one of
the flash ROMs 1004, 1008, 1015 is a destination of data
25 installing. If the destination of data installing is the
flash ROM 1004, the control flow proceeds to step S105

where the CPU 1001 erases data in a rewrite area of the flash ROM 1004. Then, the CPU 1001 sets information indicating that the data in the flash ROM was erased, and erases data at a predetermined address in the flash ROM from which information is read in above step S101. This enables the CPU to determine that the rewrite has suspended, when the data at the predetermined address is not definite. As an alternative, a flag indicating that the flash ROM is under execution of the rewrite may be set in a nonvolatile RAM.

Subsequently, the control flow proceeds to step S106 where the CPU waits for that rewrite data transmitted from the external apparatus is received through the serial interface 1011. The rewrite data transmitted from the external apparatus is divided in units of amount which can be processed in the apparatus at a time, e.g., in units of 256 bytes. Also, to shorten a transfer time, the rewrite data is transmitted in the form compressed to shorter data codes. The data compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the rewrite data is received in step S106, the control flow proceeds to step S107 where the compressed data codes are decompressed to original data.

Then, in step S108, the data decompressed in step S107 is written in the flash ROM 1004 at predetermined

addresses. It is determined in subsequent step S109 whether the rewrite data has been completely written up to the last data. If the last data is not yet written, the control flow returns to step S106 to repeat the
5 above process. If the rewrite data has been completely written up to the last data, the control flow proceeds to step S117 after clearing the aforesaid information indicating that the data in the flash ROM 1004 was erased.

10 In step S117, the check sum of the data in the flash ROM thus just rewritten is calculated to determine whether the rewrite has been normally ended. If the rewrite has been normally ended in S117, the rewrite process is ended. If the rewrite has not been normally
15 ended, the control flow proceeds to S118 where information indicating the abnormal end is left, followed by ending the rewrite process. The information indicating the abnormal end may be written in the flash ROM at a predetermined address. As an alternative, that
20 information may be written in a nonvolatile RAM.

If it is determined in above step S104 that the destination of data installing is the flash ROM 1008, the control flow proceeds to step S110 where the CPU 1001 instructs the CPU 1002 to erase data in a rewrite
25 area of the flash ROM 1008 through the dual port RAM 1007. Using the same above-described manner as taken by

the CPU 1001, the CPU 1002 sets information indicating that data in the flash ROM 1008 was erased, so that the set information can be referred to at the time of determining in above step S100 whether the rewrite of
5 the flash ROM has suspended. Subsequently, the CPU 1001 waits for that rewrite data is received through the serial interface 1011. The rewrite data transmitted from the external apparatus is divided in units of amount which can be processed in the apparatus at a time, e.g.,
10 in units of 256 bytes. Also, to shorten a transfer time, the rewrite data is transmitted in the form compressed to shorter data codes. The data compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the rewrite data is received in step
15 S110, the control flow proceeds to step S111 where the compressed data codes are decompressed to original data.

Then, in step S112, the data decompressed in step S111 is transferred to the CPU 1002 through the dual port RAM 1007. The CPU 1002 writes the data transmitted
20 from the CPU 1001 through the dual port RAM 1007 in the flash ROM 1008 at predetermined addresses. After transmitting the data in step S112, the CPU 1001 proceeds to step S113.

It is determined in step S113 whether the rewrite
25 data has been completely written up to the last data. If the last data is not yet written, the control flow

returns to step S110 to repeat the above process. If it is determined in step S113 that the rewrite data has been completely written up to the last data, the CPU 1001 notifies the CPU 1002 of the fact that the last
5 data has been written, and in response to such a notice, the CPU 1002 clears the aforesaid information indicating that the data in the flash ROM 1004 was erased. After that, the CPU 1001 proceeds to step S117 mentioned above.

If it is determined in above step S104 that the
10 destination of data installing is the flash ROM 1015, the control flow proceeds to step S114 where the CPU 1001 presents display on the control panel 300 which informs the operator of the necessity of rewriting the flash ROM 1015. Then, the CPU 1001 instructs the CPU
15 1014 to erase data in a rewrite area of the flash ROM 1015 through the dual port RAM 1013. Using the same above-described manner as taken by the CPU 1001, the CPU 1014 sets information indicating that data in the flash ROM 1015 was erased, so that the set information can be
20 referred to at the time of determining in above step S100 whether the rewrite of the flash ROM has suspended.

Subsequently, the CPU 1001 waits for that rewrite data
is received through the serial interface 1011. The rewrite data transmitted from the external apparatus is
25 divided in units of amount which can be processed in the apparatus at a time, e.g., in units of 256 bytes. Also,

to shorten a transfer time, the rewrite data is transmitted in the form compressed to shorter data codes.

The data compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the
5 rewrite data is received in step S114, the control flow proceeds to step S115.

In step S115, the received data is transferred to the CPU 1014 through the dual port RAM 1013. The CPU 1014 decompresses the data, transferred from the CPU
10 1001 through the dual port RAM 1013, to original data which is then written in the flash ROM 1015 at predetermined addresses. After transmitting the data in step S115, the CPU 1001 proceeds to step S116.

It is determined in step S116 whether the rewrite
15 data has been completely written up to the last data. If the last data is not yet written, the control flow returns to step S114 to repeat the above process. If it is determined in step S116 that the rewrite data has been completely written up to the last data, the CPU
20 1001 notifies the CPU 1014 of the fact that the last data has been written, and in response to such a notice, the CPU 1014 clears the aforesaid information indicating that the data in the flash ROM 1015 was erased. After that, the CPU 1001 proceeds to step S117 mentioned above.

25 Fig. 6 shows an example of display indicated when the flash ROM 1004 is rewritten. As illustrated, instead

of the normal display for copying setting, the control panel displays a message informing the operator of the fact that copying is disabled and the apparatus is under download of data into the flash ROM 1004. Also, when the
5 flash ROMs 1008, 1015 are rewritten, a similar message to that shown in Fig. 6 is displayed on the control panel.

[Second Embodiment]

A second embodiment of the present invention will
10 be described below in detail with reference to the drawings.

Note that a sectional structure of an image forming apparatus according to the second embodiment and a construction of a control panel of the apparatus are
15 similar to those of the image forming apparatus according to the first embodiment shown in Figs. 1 and 2; hence a description thereof is omitted here.

Fig. 8 is a block diagram showing a construction of a control system for an image forming apparatus 100
20 according to this second embodiment. In Fig. 8, reference numeral 1001 denotes a CPU for carrying out basic control of the entire apparatus. The CPU 1001 is connected, via an address bus and a data bus, to an input/output (I/O) port 1003, a flash ROM 1004 in which
25 image forming control programs are written, a work RAM 1005 for use in execution of processing, a flash ROM

1000 in which a download program is stored, and a key/display control unit 1006 for controlling display on and key inputs from a control panel 300.

In addition, the CPU 1001 includes a built-in ROM
5 1019 in which a program for transferring a control program adapted to rewrite the flash ROM 1004, etc. based on data transmitted from an external apparatus, such as a host computer (personal computer), to the flash ROM 1000 is written. Also, the CPU 1001 has a
10 plurally of operating modes which can be selectively set by an operating mode setting unit 1018. During the normal image forming operation, the CPU 1001 is set to the mode in which the built-in ROM 1019 is not effective.

In such a control system, the CPU 1001 executes
15 input/output control successively in accordance with the contents of the flash ROM 1004 for carrying out the image forming operation, and also receives key-input information from the control panel 300 through the key/display control unit 1006. Further, the CPU 1001
20 instructs the key/-display control unit 1006 to indicate the operating states of the apparatus 100, the operating mode set in response to the key input, etc., thereby controlling display on the control panel 300.

The image forming apparatus 100 also includes
25 another CPU 1002 for executing control of operations of various motors, heaters and so on. The operation control

of the various motors, etc. is performed in accordance with instructions from the CPU 1002. The CPU 1002 is connected, via an address bus and a data bus, to a flash ROM 1008 in which control programs are written, a work
5 RAM 1009 for use in execution of processing, and an input/-output (I/O) port 1010. Various control motors, etc. are connected to the input/output (I/O) port 1010 as needed. Additionally, the CPU 1002 executes data communication with the CPU 1001 through a dual port RAM
10 1007 serving as interface means.

Further, the CPU 1001 executes data communication with a CPU 1014 for converting data, such as PDL codes, which is transmitted from the external apparatus, into an image signal through a dual port RAM 1013 serving as
15 interface means. Upon receiving the PDL codes from the external apparatus, the CPU 1014 converts the received PDL codes into image data, requests image forming processing to the CPU 1001, which is control means for an image forming section, through the dual port RAM 1013,
20 and then outputs the image data to the printer section 2 in synch with control of the CPU 1001. The CPU 1014 is connected, via an address bus and a data bus, to a flash ROM 1015 in which control programs are written, a work RAM 1016 for use in execution of processing and for
25 developing the PDL codes into the image data, and an input/output (I/O) port 1017. Note that the input/output

port 1017 is connected to a control port through which data, such as PDL codes, is received from the external apparatus.

In addition, the CPU 1001 executes data
5 communication with a not-shown external apparatus, such as a personal computer, through a serial interface (I/F) 1011. The CPU 1001 also analyzes data received from the external apparatus through the serial interface 1011, and transmits data in the apparatus 100 to the external
10 apparatus in accordance with an analyzed result. Further, the CPU 1001 performs rewrite of data in the flash ROMs 1004, 1008, 1015 based on data received from the external apparatus.

A process for rewriting data in the flash ROMs 1004,
15 1008, 1015 will be described below. Fig. 9 is a flowchart showing the data rewrite process in each flash ROM according to this second embodiment. This process is performed by the CPUs 1001, 1002, 1014 when a power supply is turned on in a state where the built-in ROM
20 1019 is made effective. More specifically, when the data rewrite process is desired, this process is started and executed as described below by operating a dip switch or the like in a state where the power supply is not turned on, thereby establishing a mode in which the built-in
25 ROM 1019 is effective, and thereafter turning on the power supply. In this connection, Fig. 11A shows a

memory map used in the normal image forming operation,
i.e., a memory map used for the execution of processing
in the mode in which the built-in ROM 1019 is made not
effective, and Fig. 11B shows a memory map used in the
5 rewrite control operation, i.e., a memory map used for
the execution of processing in the mode in which the
built-in ROM 1019 is made effective. In other words, Fig.
11A shows a memory map used in a normal process to carry
out the operation of a copying machine or a printer, and
10 Fig. 11B shows a memory map used in a download process.

As will be seen from Figs. 11A and 11B, part of the
memory area assigned to the flash ROM in the mode in
which the built-in ROM 1019 is not effective, is
assigned to the built-in ROM 1019 in the mode in which
15 the built-in ROM 1019 is effective. In the illustrated
example, addresses from 000000h to 010000h represent an
area shared by both the modes. Further, the shared area
includes in its part a normal interrupt vector area in
the mode in which the built-in ROM 1019 is not effective,
20 and a transfer/-download vector area in the mode in
which the built-in ROM 1019 is effective. Stated
otherwise, the same memory space is occupied by the
flash ROM during the image forming operation and by the
built-in ROM during the rewrite control operation, thus
25 enabling a vector area dedicated for the rewrite control
to be ensured.

The reason why part of the shared memory area is provided as the transfer/download vector area in the illustrated example is that the memory is used upon an interrupt substantially in the same range for transfer
5 to the flash ROM 1000, and therefore a common vector area can be used in both transfer and download.

Returning to Fig. 9, when execution of the rewrite control program is started, i.e., when in this embodiment the power supply is turned on in the mode in
10 which the built-in ROM is effective, the operator is informed of the fact that the apparatus is under the rewrite process and the image forming operation cannot be effected.

First, in step S200, it is determined in accordance
15 with information described below whether the rewrite or rewrite process of each of the flash ROMs 1004, 1008, 1015 has suspended or failed. If the rewrite of the flash ROMs 1008, 1015 has suspended, information indicating the suspension is noticed from the CPUs 1002,
20 1014 to the CPU 1001 through the dual port RAMs 1007, 1013. When the rewrite is determined as having suspended in response to the above notice, this means that the control data in each flash ROM is in a state not ensured.

Therefore, the control flow proceeds to step S203
25 described later for executing a rewrite program transfer mode. Incidentally, how to determine whether the rewrite

of the flash ROM 1004 has suspended will be described later.

On the other hand, if it is determined in step S200 that the rewrite has not suspended, the control flow
5 proceeds to step S201 to read information in the flash ROM which is a destination of data installing in an install mode. In subsequent step S202, the read version is compared with the version to be newly installed. If
10 both the versions are in match with each other, there is no need of installing data. Therefore, the CPU indicates the end of the install mode to the operator, and also prompts the operator to turn off the power supply and to set the mode back to the mode in which the built-in ROM is not effective. However, if both the versions are not
15 in match with each other in step S202, the control flow proceeds to step S203 to execute the rewrite program transfer mode.

In step S203, in accordance with the contents of the built-in ROM 1019, a rewrite program is transferred
20 to the flash ROM 1000 based on data transmitted from the external apparatus. After that, the control flow proceeds to step S204 to execute a flash ROM rewrite mode.

Details of control in above step S203 will now be
25 described with reference to a flowchart of Fig. 10. When the rewrite program transfer mode is started, the

contents of information in the flash ROM are first read
in step S2031 for comparison of the stored version with
the version to be installed. If both the versions are in
match with each other, the control flow proceeds to step
5 S204 after informing the external apparatus of the fact
that there is no need of transferring the rewrite
program. If both the versions are not in match with each
other, the control flow proceeds to step S2032 to erase
data in the flash ROM 1000. In subsequent step S2033,
10 the CPU 1001 waits for reception of data from the
external apparatus. After receiving the data, If both
the versions are in match with each other, the control
flow proceeds to step S2034 to write the data in
accordance with signals received by the flash ROM 1000.
15 Then, it is determined in step S2035 whether the
received data is the last data to be transferred. If the
received data is the last data, the control flow
proceeds to step S204 to execute the flash ROM rewrite
mode. If the received data is not the last data, the
20 control flow returns to step S2033 to repeat the above
process.

In this connection, the built-in ROM 1019 stores
therein a control program for transferring the rewrite
program to the flash ROM 1000. Comparing with the case
25 of directly storing, in the built-in ROM 1019, the
rewrite program for rewriting the image forming control

programs stored in a plurality of memory media, however, it is expected that the amount of program data to be stored in the built-in ROM 1019 is reduced considerably as a result of the above process of this embodiment.

- 5 Correspondingly, it is also expected that a probability of bugs being present in the built-in ROM 1019 is reduced considerably.

Next, it is determined in step S204 which one of the flash ROMs 1004, 1008, 1015 is a data destination of installing. If the destination of data installing is the
10 flash ROM 1004, the control flow proceeds to step S205 where the CPU 1001 erases data in a rewrite area of the flash ROM 1004. Then, the CPU 1001 sets information indicating that the data in the flash ROM was erased,
15 and erases data at a predetermined address in the flash ROM from which information is read in above step S201. This enables the CPU to determine that the rewrite has suspended, when the data at the predetermined address is not definite. As an alternative, a flag indicating that
20 the flash ROM is under execution of the rewrite may be set in a nonvolatile RAM.

Subsequently, the control flow proceeds to step S206 where the CPU waits for that rewrite data transmitted from the external apparatus is received
25 through the serial interface 1011. The rewrite data transmitted from the external apparatus is divided in

units of amount which can be processed in the apparatus at a time, e.g., in units of 256 bytes. Also, to shorten a transfer time, the rewrite data is transmitted in the form compressed to shorter data codes. The data

5 compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the rewrite data is received in step S206, the control flow proceeds to step S207 where the compressed data codes are decompressed to original data.

10 Then, in step S208, the data decompressed in step S207 is written in the flash ROM 1004 at predetermined addresses. It is determined in subsequent step S209 whether the rewrite data has been completely written up to the last data. If the last data is not yet written,
15 the control flow returns to step S206 to repeat the above process. If the rewrite data has been completely written up to the last data, the control flow proceeds to step S217 after clearing the aforesaid information indicating that the data in the flash ROM 1004 was
20 erased.

In step S217, the check sum of the data in the flash ROM thus just rewritten is calculated to determine whether the rewrite has been normally ended. If the rewrite has been normally ended in S217, the rewrite
25 process is ended. If the rewrite has not been normally ended, the control flow proceeds to S218 where

information indicating the abnormal end is left,
followed by ending the rewrite process. The information
indicating the abnormal end may be written in the flash
ROM at a predetermined address. As an alternative, that
5 information may be written in a nonvolatile RAM.

If it is determined in above step S204 that the
destination of data installing is the flash ROM 1008,
the control flow proceeds to step S210 where the CPU
1001 instructs the CPU 1002 to erase data in a rewrite
10 area of the flash ROM 1008 through the dual port RAM
1007. Using the same above-described manner as taken by
the CPU 1001, the CPU 1002 sets information indicating
that data in the flash ROM 1008 was erased, so that the
set information can be referred to at the time of
15 determining in above step S200 whether the rewrite of
the flash ROM has suspended. Subsequently, the CPU 1001
waits for that rewrite data is received through the
serial interface 1011. The rewrite data transmitted from
the external apparatus is divided in units of amount
20 which can be processed in the apparatus at a time, e.g.,
in units of 256 bytes. Also, to shorten a transfer time,
the rewrite data is transmitted in the form compressed
to shorter data codes. The data compression is performed
in accordance with Lempel-Ziv coding, for example.
25 Thereafter, when the rewrite data is received in step
S210, the control flow proceeds to step S211 where the

compressed data codes are decompressed to original data.

Then, in step S212, the data decompressed in step S211 is transferred to the CPU 1002 through the dual port RAM 1007. The CPU 1002 writes the data transmitted
5 from the CPU 1001 through the dual port RAM 1007 in the flash ROM 1008 at predetermined addresses. After transmitting the data in step S212, the CPU 1001 proceeds to step S213.

It is determined in step S213 whether the rewrite
10 data has been completely written up to the last data. If the last data is not yet written, the control flow returns to step S210 to repeat the above process. If it is determined in step S213 that the rewrite data has been completely written up to the last data, the CPU
15 1001 notifies the CPU 1002 of the fact that the last data has been written, and in response to such a notice, the CPU 1002 clears the aforesaid information indicating that the data in the flash ROM 1008 was erased. After that, the CPU 1001 proceeds to step S217 mentioned above.

20 If it is determined in above step S204 that the destination of data installing is the flash ROM 1015, the control flow proceeds to step S214 where the CPU 1001 presents display on the control panel 300 which informs the operator of the necessity of rewriting the
25 flash ROM 1015. Then, the CPU 1001 instructs the CPU 1014 to erase data in a rewrite area of the flash ROM

1015 through the dual port RAM 1013. Using the same
above-described manner as taken by the CPU 1001, the CPU
1014 sets information indicating that data in the flash
ROM 1015 was erased, so that the set information can be
5 referred to at the time of determining in above step
S200 whether the rewrite of the flash ROM has suspended.

Subsequently, the CPU 1001 waits for that rewrite data
is received through the serial interface 1011. The
rewrite data transmitted from the external apparatus is
10 divided in units of amount which can be processed in the
apparatus at a time, e.g., in units of 256 bytes. Also,
to shorten a transfer time, the rewrite data is
transmitted in the form compressed to shorter data codes.

The data compression is performed in accordance with
15 Lempel-Ziv coding, for example. Thereafter, when the
rewrite data is received in step S214, the control flow
proceeds to step S215.

In step S215, the received data is transferred to
the CPU 1014 through the dual port RAM 1013. The CPU
20 1014 decompresses the data, transferred from the CPU
1001 through the dual port RAM 1013, to original data
which is then written in the flash ROM 1015 at
predetermined addresses. After transmitting the data in
step S215, the CPU 1001 proceeds to step S216.

25 It is determined in step S216 whether the rewrite
data has been completely written up to the last data. If

the last data is not yet written, the control flow returns to step S214 to repeat the above process. If it is determined in step S216 that the rewrite data has been completely written up to the last data, the CPU
5 1001 notifies the CPU 1014 of the fact that the last data has been written, and in response to such a notice, the CPU 1014 clears the aforesaid information indicating that the data in the flash ROM 1015 was erased. After that, the CPU 1001 proceeds to step S217 mentioned above.

10 Also in this second embodiment, as with the first embodiment, instead of the normal display for copying setting, the control panel displays a message, as shown in Fig. 6, informing the operator of the fact that copying is disabled and the apparatus is under download
15 of data into the flash ROM 1004. Further, when the flash ROMs 1008, 1015 are rewritten, a similar message is displayed on the control panel.

While the second embodiment has been described as presenting display, which indicates that the flash ROM
20 1004, 1008 or 1015 is under execution of the rewrite, in the display section 301 of the control panel 300, an indicator lamp showing a state under execution of the rewrite when blinked, for example, may be disposed on the control panel 300 or any position on the image
25 forming apparatus that can be easily viewed by the operator for confirmation. In such a case, for example,

an LED is disposed on the control panel 300 and is
turned off in a normal state. Then, the LED is turned on
when the rewrite process of the flash ROM is failed.
Under execution of the rewrite process, the LED is
5 blinked.

Moreover, instead of using the display means of the
image forming apparatus, such as the control panel 300,
to display the state of the flash ROM, such display may
be presented on a host connected for rewriting the flash
10 ROM. In this case, instead of the process for presenting
display on the control panel 300, the CPU 1001 informs
the necessity of rewriting the flash ROM to the host
through the serial interface 1011 in the form of RS-232C.
The host then displays the informed notice on its image
15 screen.

While the flash ROM is used as rewritable memory
means in the second embodiment, any other suitable
memory means such as a hard disk can also be similarly
employed instead of the flash ROM.

20 According to the second embodiment, as described
above, a program for transferring a download program is
built in a ROM or the like, and control programs for the
image forming operation are rewritten by executing the
transferred download program. Information stored in a
25 flash ROM slightly tends to be unstable, and the
contents of control programs for the image forming

A third embodiment of the present invention will be described below in detail with reference to the drawings.

Note that a sectional structure of an image forming apparatus according to the third embodiment and a
5 construction of a control panel of the apparatus are similar to those of the image forming apparatus according to the second embodiment shown in Figs. 1 and 2; hence a description thereof is omitted here.

Fig. 12 is a block diagram showing a construction
10 of a control system for an image forming apparatus 100 according to this third embodiment. In Fig. 12, reference numeral 1001 denotes a CPU for carrying out basic control of the entire apparatus. The CPU 1001 is connected, via an address bus and a data bus, to an
15 input/output (I/O) port 1003, a flash ROM(a) 1004 in which image forming control programs are written, a selector 1018 for selecting either a download ROM 1021 in which a download program is written or a flash ROM(b) 1020 which includes an interrupt vector area for use in
20 the image forming operation, a work RAM 1005 for use in execution of processing, and a key/display control unit 1006 for controlling display on and key inputs from a control panel 300.

In addition, the CPU 1001 includes a switch 1019
25 changed over so that either the download ROM 1021 in which the download program is written or the flash

ROM(b) 1020 which includes the interrupt vector area for use in the image forming operation is made effective through the selector 1018. The change-over of the switch 1019 is set to make effective the flash ROM(b) 1020

5 which includes the interrupt vector area, when the apparatus is under the normal image forming operation. This change-over control will be described later.

In such a control system, the CPU 1001 executes input/output control successively in accordance with the
10 contents of the flash ROM for carrying out the image forming operation, and also receives key-input information from the control panel 300 through the key/display control unit 1006. Further, the CPU 1001 instructs the key/display control unit 1006 to indicate
15 the operating states of the apparatus 100, the operating mode set in response to the key input, etc., thereby controlling display on the control panel 300.

The image forming apparatus 100 also includes another CPU 1002 for executing control of operations of
20 various motors, heaters and so on. The operation control of the various motors, etc. is performed in accordance with instructions from the CPU 1002. The CPU 1002 is connected, via an address bus and a data bus, to a flash ROM 1008 in which control programs are written, a work
25 RAM 1009 for use in execution of processing, and an input/-output (I/O) port 1010. Various control motors,

etc. are connected to the input/output (I/O) port 1010 as needed. Additionally, the CPU 1002 executes data communication with the CPU 1001 through a dual port RAM 1007 serving as interface means.

5 Further, the CPU 1001 executes data communication with a CPU 1014 for converting data, such as PDL codes, which is transmitted from an external apparatus such as a host computer (personal computer), into an image signal through a dual port RAM 1013 serving as interface
10 means. Upon receiving the PDL codes from the external apparatus, the CPU 1014 converts the received PDL codes into image data, requests image forming processing to the CPU 1001, which is control means for an image forming section, through the dual port RAM 1013, and
15 then outputs the image data to the printer section 2 in synch with control of the CPU 1001. The CPU 1014 is connected, via an address bus and a data bus, to a flash ROM 1015 in which control programs are written, a work RAM 1016 for use in execution of processing and for
20 developing the PDL codes into the image data, and an input/output (I/O) port 1017. Note that the input/output (I/O) port 1017 is connected to a control port through which data, such as PDL codes, is received from the external apparatus.

25 In addition, the CPU 1001 executes data communication with a not-shown external apparatus, such

as a personal computer, through a serial interface (I/F) 1011. The CPU 1001 also analyzes data received from the external apparatus through the serial interface 1011, and transmits data in the apparatus 100 to the external apparatus in accordance with an analyzed result. Further, the CPU 1001 performs rewrite of data in the flash ROMs 1004, 1008, 1015 based on data received from the external apparatus.

A process for rewriting data in the flash ROMs 1004, 1008, 1015 will be described below. Fig. 13 is a flowchart showing the data rewrite process in each flash ROM according to this third embodiment. This process is performed by the CPU 1001 when a power supply is turned on in a state where the download ROM 1021 is made effective with operation of the switch 1019. More specifically, when the data rewrite process is desired, this process is started and executed as described below by setting the switch 1019 to make effective the download ROM 1021 in a state where the power supply is not turned on, and thereafter turning on the power supply. In this connection, Fig. 14A shows a memory map used in the normal image forming operation, i.e., a memory map used for the execution of processing when the flash ROM(b) 1020 is made effective, and Fig. 14B shows a memory map used in the rewrite control operation, i.e., a memory map used for the execution of processing when

the download ROM 1021 is made effective.

As will be seen from Figs. 14A and 14B, part of the memory area assigned to the flash ROM(b) 1020 during the normal image forming operation, is assigned to the

5 download ROM 1021 during the rewrite control operation.

In the illustrated example, addresses from 000000h to 010000h represent an area shared by both the ROMs.

Further, the shared area includes in its part a normal interrupt vector area during the normal image forming
10 operation, and a download vector area during the rewrite control operation. Stated otherwise, the same memory space is occupied by the flash ROM during the image forming operation and by the download ROM during the rewrite control operation, thus enabling a vector area
15 dedicated for the rewrite control to be ensured.

Returning to Fig. 13, when execution of the rewrite control program is started, i.e., when in this embodiment the power supply is turned on in a state where the switch 1019 is changed over to make the
20 download ROM effective, the operator is informed of the fact that the apparatus is under the rewrite process and the image forming operation cannot be effected.

First, in step S301, it is determined in accordance with information described below whether the rewrite or
25 rewrite process of each of the flash ROMs 1004, 1008, 1015 has suspended or failed. If the rewrite of the

03615194 121833

flash ROMs 1008, 1015 has suspended, information
indicating the suspension is noticed from the CPUs 1002,
1014 to the CPU 1001 through the dual port RAMs 1007,
1013. When the rewrite is determined as having suspended
5 in response to the above notice, this means that the
control data in each flash ROM is in a state not ensured.
Therefore, the control flow proceeds to step S304
described later for executing a flash ROM rewrite mode.
Incidentally, how to determine whether the rewrite of
10 the flash ROM 1004 has suspended will be described later.
On the other hand, if it is determined in step S301
that the rewrite has not suspended, the control flow
proceeds to step S302 to read information in the flash
ROM which is a destination of data installing in an
15 install mode. In subsequent step S303, the read version
is compared with the version to be newly installed. If
both the versions are in match with each other, there is
no need of installing data. Therefore, the CPU indicates
the end of the install mode to the operator, and also
20 prompts the operator to turn off the power supply and to
change over the switch 1019 for setting the flash ROM(b)
to be effective. However, if both the versions are not
in match with each other in step S303, the control flow
proceeds to step S304 to execute the flash ROM rewrite
25 mode.

Next, it is determined in step S304 which one of

the flash ROMs 1004, 1008, 1015 is a destination of data installing. If the destination of data installing is the flash ROM 1004, the control flow proceeds to step S305 where the CPU 1001 erases data in a rewrite area of the
5 flash ROM 1004. Then, the CPU 1001 sets information indicating that the data in the flash ROM was erased, and erases data at a predetermined address in the flash ROM from which information is read in above step S301. This enables the CPU to determine that the rewrite has
10 suspended, when the data at the predetermined address is not definite. As an alternative, a flag indicating that the flash ROM is under execution of the rewrite may be set in a nonvolatile RAM.

Subsequently, the control flow proceeds to step
15 S306 where the CPU waits for that rewrite data transmitted from the external apparatus is received through the serial interface 1011. The rewrite data transmitted from the external apparatus is divided in units of amount which can be processed in the apparatus
20 at a time, e.g., in units of 256 bytes. Also, to shorten a transfer time, the rewrite data is transmitted in the form compressed to shorter data codes. The data compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the rewrite data
25 is received in step S306, the control flow proceeds to step S307 where the compressed data codes are

decompressed to original data.

Then, in step S308, the data decompressed in step S307 is written in the flash ROM 1004 at predetermined addresses. It is determined in subsequent step S309
5 whether the rewrite data has been completely written up to the last data. If the last data is not yet written, the control flow returns to step S306 to repeat the above process. If the rewrite data has been completely written up to the last data, the control flow proceeds
10 to step S317 after clearing the aforesaid information indicating that the data in the flash ROM 1004 was erased.

In step S317, the check sum of the data in the flash ROM thus just rewritten is calculated to determine
15 whether the rewrite has been normally ended. If the rewrite has been normally ended in S317, the rewrite process is ended. If the rewrite has not been normally ended, the control flow proceeds to S318 where information indicating the abnormal end is left,
20 followed by ending the rewrite process. The information indicating the abnormal end may be written in the flash ROM at a predetermined address. As an alternative, that information may be written in a nonvolatile RAM.

If it is determined in above step S304 that the
25 destination of data installing is the flash ROM 1008, the control flow proceeds to step S310 where the CPU

1001 instructs the CPU 1002 to erase data in a rewrite
area of the flash ROM 1008 through the dual port RAM
1007. Using the same above-described manner as taken by
the CPU 1001, the CPU 1002 sets information indicating
5 that data in the flash ROM 1008 was erased, so that the
set information can be referred to at the time of
determining in above step S301 whether the rewrite of
the flash ROM has suspended. Subsequently, the CPU 1001
waits for that rewrite data is received through the
10 serial interface 1011. The rewrite data transmitted from
the external apparatus is divided in units of amount
which can be processed in the apparatus at a time, e.g.,
in units of 256 bytes. Also, to shorten a transfer time,
the rewrite data is transmitted in the form compressed
15 to shorter data codes. The data compression is performed
in accordance with Lempel-Ziv coding, for example.
Thereafter, when the rewrite data is received in step
S310, the control flow proceeds to step S311 where the
compressed data codes are decompressed to original data.
20 Then, in step S312, the data decompressed in step
S311 is transferred to the CPU 1002 through the dual
port RAM 1007. The CPU 1002 writes the data transmitted
from the CPU 1001 through the dual port RAM 1007 in the
flash ROM 1008 at predetermined addresses. After
25 transmitting the data in step S312, the CPU 1001
proceeds to step S313.

It is determined in step S313 whether the rewrite data has been completely written up to the last data. If the last data is not yet written, the control flow returns to step S310 to repeat the above process. If it is determined in step S313 that the rewrite data has been completely written up to the last data, the CPU 1001 notifies the CPU 1002 of the fact that the last data has been written, and in response to such a notice, the CPU 1002 clears the aforesaid information indicating that the data in the flash ROM 1008 was erased. After that, the CPU 1001 proceeds to step S317 mentioned above.

If it is determined in above step S304 that the destination of data installing is the flash ROM 1015, the control flow proceeds to step S314 where the CPU 1001 presents display on the control panel 300 which informs the operator of the necessity of rewriting the flash ROM 1015. Then, the CPU 1001 instructs the CPU 1014 to erase data in a rewrite area of the flash ROM 1015 through the dual port RAM 1013. Using the same above-described manner as taken by the CPU 1001, the CPU 1014 sets information indicating that data in the flash ROM 1015 was erased, so that the set information can be referred to at the time of determining in above step S301 whether the rewrite of the flash ROM has suspended. Subsequently, the CPU 1001 waits for that rewrite data is received through the serial interface 1011. The

rewrite data transmitted from the external apparatus is divided in units of amount which can be processed in the apparatus at a time, e.g., in units of 256 bytes. Also, to shorten a transfer time, the rewrite data is

5 transmitted in the form compressed to shorter data codes.

The data compression is performed in accordance with Lempel-Ziv coding, for example. Thereafter, when the rewrite data is received in step S314, the control flow proceeds to step S315.

10 In step S315, the received data is transferred to the CPU 1014 through the dual port RAM 1013. The CPU 1014 decompresses the data, transferred from the CPU 1001 through the dual port RAM 1013, to original data which is then written in the flash ROM 1015 at
15 predetermined addresses. After transmitting the data in step S315, the CPU 1001 proceeds to step S316.

It is determined in step S316 whether the rewrite data has been completely written up to the last data. If the last data is not yet written, the control flow
20 returns to step S314 to repeat the above process. If it is determined in step S316 that the rewrite data has been completely written up to the last data, the CPU 1001 notifies the CPU 1014 of the fact that the last data has been written, and in response to such a notice,
25 the CPU 1014 clears the aforesaid information indicating that the data in the flash ROM 1015 was erased. After

that, the CPU 1001 proceeds to step S317 mentioned above.

Also in this third embodiment, as with the first embodiment, instead of the normal display for copying setting, the control panel displays a message, as shown
5 in Fig. 6, informing the operator of the fact that copying is disabled and the apparatus is under download of data into the flash ROM 1004. Further, when the flash ROMs 1008, 1015 are rewritten, a similar message is displayed on the control panel.

10 According to the third embodiment described above, the rewrite process can be performed in a manner of interrupt. Therefore, when data is transmitted, the rewrite process can be started as an interrupt process at once and a processing time required for rewrite can
15 be shortened. Information stored in a flash ROM slightly tends to be unstable. This means that if the contents of a flash ROM have become not normal, the rewrite control program may be impossible to execute when it is stored in the same flash ROM. With the third embodiment, since
20 the rewrite program is stored in a ROM, it can be executed at any time regardless of the contents of a flash ROM, and even if the contents of the flash ROM have become not normal, they can be restored to a normal state.

25 As described hereinabove, according to the first to third embodiments, since a control program for just

transferring the download program to a RAM or a flash ROM is stored in a ROM or the like, and the rewrite process is performed by executing the transferred download program, it is possible to minimize the amount of program data to be stored in the ROM or the like, and to improve the reliability.

Further, according to the first to third embodiments, the rewrite process can be achieved at a high speed because the interrupt is allowed in not only the image forming process, but also the rewrite process. [Fourth Embodiment]

A fourth embodiment of the present invention will be described with reference to Figs. 15 and 16.

Fig. 16 is a block diagram showing a construction of an image forming apparatus to which a control program writing method according to a fourth embodiment can be applied.

An image forming apparatus 1200 of this fourth embodiment is constructed of, e.g., a copying machine or a printer. The image forming apparatus 1200 primarily comprises a CPU 1201, a RAM 1202, a flash ROM 1203, a data input unit 1204, an EPROM 1205, a bus system connecting those components with one another, and a dip switch 1206.

The CPU 1201 is a central processing unit for executing arithmetic and control operations to perform

control of the entire apparatus. The RAM 1202 is a memory medium for temporarily storing data, the flash ROM 1203 is a ROM for storing control programs, fixed values, etc. which are used processing executed by the CPU 201, and the EPROM 1205 is a ROM for storing a program, such as a loading program, which is not updated.

The data input unit 1204 is an interface for connection between those components and a host computer 1208.

The dip switch 1206 is a switch for selecting whether the apparatus is operated in a normal manner or a control program is down-loaded. The dip switch 1206 is changed over to select a program to be started up upon power-on.

Fig. 15 is a flowchart showing a control program write process according to the fourth embodiment.

After connecting the host computer 1208 and the data input unit 1204 to each other (step S401), a power supply of the image forming apparatus 1200 is turned on (step S402).

Upon power-on of the apparatus, the CPU 1201 determines depending on the state of the dip switch 1206 whether a normal operating mode or a control program download mode is selected (step S403).

If the dip switch 1206 selects the normal operating mode, the CPU 1201 starts up the control programs stored in the flash ROM 1203 after power-on (step S404).

If the dip switch 1206 selects the control program download mode, the CPU 1201 starts up the loading program stored in the EPROM 1205 after power-on (step S405).

5 When the loading program is started up, the CPU 1201 is brought into a state ready for receiving a rewrite program from the host computer 1208. The host computer 1208 is then operated to transmit the rewrite program which is received by the data input unit 1204.

10 At this time, address information for the RAM 1202 is added to the rewrite program in the host computer 1208. The loading program reads the address information and loads the rewrite program at predetermined addresses of the RAM 1202 (step S406). In other words, the CPU
15 1201 loads the data, which has been received by the data input unit 1204, in the RAM 1202 in accordance with the address information added to the rewrite program. After the rewrite program has been received and loaded, the CPU 1201 starts up the rewrite program loaded in the RAM
20 1202, and is brought into a state ready for receiving the control programs from the host computer 1208 (step S407). The host computer 1208 is then operated to transmit the control programs which are received by the data input unit 1204.

25 At this time, address information for the flash ROM 1203 is added to the control programs, and the rewrite

program can be designed in match with the control programs. Therefore, the control programs can be designed freely without suffering restrictions from the rewrite program.

5 Moreover, since the rewrite program is down-loaded along with the address information for the RAM 1202 added to it and the control programs are down-loaded along with the address information for the flash ROM 1203 added to it, there is obtained such an advantage
10 that even when the program allocated addresses are changed in terms of hardware, the rewrite program and the control programs can be written without being affected by the address change.

[Fifth Embodiment]

15 A fifth embodiment of the present invention will be described below with reference to Fig. 17.

 In the above fourth embodiment, the control program download mode and the normal operating mode are changed over depending on the state of the dip switch 1206. On
20 the other hand, this fifth embodiment is constructed to change over the control program download mode and the normal operating mode in response to a command from the host computer 1208.

 The other construction and operation of the fifth
25 embodiment are similar to those of the fourth embodiment; hence a description thereof is omitted here.

Namely, a control program writing method according to this fifth embodiment can be applied to not only the apparatus shown in Fig. 16 with no need of modifying the apparatus, but also an apparatus in which the dip switch
5 1206 among the components the apparatus shown in Fig. 16 is eliminated.

Fig. 17 is a flowchart showing a control program write process according to the fifth embodiment.

After connecting the host computer 1208 and the
10 data input unit 1204 to each other (step S501), the power supply of the image forming apparatus 1200 is turned on (step S502). Upon power-on of the apparatus, the CPU 1201 starts up the loading program stored in the EPROM 1205 after confirming the connection with the host
15 computer 1208, and it is brought into a state ready for receiving data from the host computer 1208 (step S503). When the first data is received (step S504), the CPU 1201 confirms whether the received data is an update command (step S505). If the received data is the update
20 command, the CPU 1201 executes the processing subsequent to step S507. If the received data is not the update command, the CPU 1201 starts up the control programs in step S506.

If the received data is the update command, the CPU
25 1201 is brought into a state ready for receiving the write program from the host computer 1208. The host

computer 1208 is then operated to transmit the rewrite program which is received by the data input unit 1204.

At this time, address information for the RAM 1202 is added to the rewrite program.

5 Subsequently, the CPU 1201 loads the data, which has been received by the data input unit 1204, in the RAM 1202 in accordance with the address information added to the rewrite program (step S507). After the rewrite program has been received and loaded, the CPU
10 1201 starts up the rewrite program loaded in the RAM 1202, and is brought into a state ready for receiving the control programs from the host computer 1208 (step S508). The host computer 1208 is then operated to transmit the control programs which are received by the
15 data input unit 1204.

At this time, address information for the flash ROM 1203 is added to the control programs.

Then, in accordance with the address information added to the rewrite program, the CPU 1201 specifies an
20 area on the flash ROM 1203 to be rewritten, and then erases data in the specified area. After the erasing, the CPU 1201 loads the data, which has been received by the data input unit 1204, in the specified area of the flash ROM 1203 in accordance with the added address
25 information (step S509).

With the process described above, in addition to

the advantages obtainable with the above fourth embodiment, an advantage of needing no dip switch and achieving a simpler hardware construction of the apparatus can be provided by this fifth embodiment.

5 Thus, according to the fourth and fifth embodiments, it is possible to efficiently down-load the control programs and design the control programs with a large flexibility. Further, even when the program allocated addresses are changed in terms of hardware, the download
10 control program (rewrite program) which is installed in the apparatus and download applications which are operated on the host computer can be used as they are without being affected by the address change.

[Other Embodiments]

15 The present invention may be applied to not only a system comprising a plurality of equipment (e.g., a host computer, an interface unit, a reader and a printer), but also an apparatus comprising a single piece of equipment (e.g., a copying machine and a facsimile).

20 While the above embodiments have been described as supplying the rewrite program and the control programs from the host computer, it is possible to achieve the object of the present invention in other forms. Thus, the present invention can also be of course implemented
25 by supplying a memory medium, which stores therein program codes of software for realizing the function of

any of the above-described embodiments, to a system or apparatus, and by causing a computer (CPU or MPU) in the system or apparatus to read and execute the program codes, such as the rewrite program and the control
5 programs, stored in the memory medium.

In such a case, the program codes read out of the memory medium serve in themselves to realize the function of any of the above-described embodiments, and therefore the memory medium storing the program codes
10 constitutes the present invention.

Memory media for storing the program codes may be, e.g., floppy disks, hard disks, optical disks, magneto-optical disks, CD-ROMs, CD-Rs, magnetic tapes, nonvolatile memory cards, and ROMs.

15 Also, the function of any of the above-described embodiments is realized by not only a computer executing the program codes read by it, but also the case where an OS (Operating System) or the like, which is working on the computer, executes a part or the whole of actual
20 processing in accordance with instructions of the program codes. It is thus a matter of course that the above latter case is also involved in modified embodiments of the present invention.

Further, it is needless to say that the present
25 invention involves such a case where the program codes read out of the memory medium are stored in a memory

built in a function add-in board of a computer or a
function add-in unit connected to the computer, and a
CPU or the like incorporated in the function add-in
board or unit then executes a part or the whole of the
5 actual processing in accordance with instructions from
the program codes, thereby realizing the function of any
of the above-described embodiments.

Where the present invention is applied to the
above-mentioned memory medium, programs codes
10 corresponding to the flowchart described above in
connection with any of the embodiments are stored in the
memory medium.

The present invention has been described above in
connection with several preferred embodiments, the
15 present invention is not limited to the illustrated
embodiments and may be modified in various ways within
the scope defined in the appended Claims.